

MMM		MMM	TTTTTTT	TTTTTTT	HHH	HHH	RRRRRRRRRRRRR		TTTTTTTTTTTTTTTTT	LLL
MMM		MMM	TTTTTTT	TTTTTTT	HHH	HHH	RRRRRRRRRRRRR		TTTTTTTTTTTTTTTTT	LLL
MMM		MMM	TTTTTTT	TTTTTTT	HHH	HHH	RRRRRRRRRRRRR		TTTTTTTTTTTTTTTTT	LLL
MMMMMM	MMMMMM		TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMMMMM	MMMMMM		TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMMMMM	MMMMMM		TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM	MMM	MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM	MMM	MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM	MMM	MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHHHHHHHHHHHHHHHH		RRRRRRRRRRRRR		TTT	LLL
MMM		MMM	TTT		HHHHHHHHHHHHHHHHH		RRRRRRRRRRRRR		TTT	LLL
MMM		MMM	TTT		HHHHHHHHHHHHHHHHH		RRRRRRRRRRRRR		TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLLLLLLLLLLLLLLLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLLLLLLLLLLLLLLLL
MMM		MMM	TTT		HHH	HHH	RRR	RRR	TTT	LLLLLLLLLLLLLLLLL

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

```

MM      MM      TTTTTTTTTT  HH      HH  DDDDDDDD  SSSSSSSS  IIIIII  NN      NN  HH      HH
MM      MM      TTTTTTTTTT  HH      HH  DDDDDDDD  SSSSSSSS  IIIIII  NN      NN  HH      HH
MMM     MMM     TT          HH      HH  DD      DD  SS      SS      II      NN      NN  HH      HH
MMM     MMM     TT          HH      HH  DD      DD  SS      SS      II      NN      NN  HH      HH
MM      MM      TT          HH      HH  DD      DD  SS      SS      II      NNNN   NN  HH      HH
MM      MM      TT          HH      HH  DD      DD  SS      SS      II      NNNN   NN  HH      HH
MM      MM      TT          HHHHHHHHHH  DD      DD  SSSSSS  SSSSSS  II      NN      NN  HHHHHHHHHH
MM      MM      TT          HHHHHHHHHH  DD      DD  SSSSSS  SSSSSS  II      NN      NN  HHHHHHHHHH
MM      MM      TT          HH      HH  DD      DD  SS      SS      II      NN      NNNN  HH      HH
MM      MM      TT          HH      HH  DD      DD  SS      SS      II      NN      NNNN  HH      HH
MM      MM      TT          HH      HH  DD      DD  SS      SS      II      NN      NN  HH      HH
MM      MM      TT          HH      HH  DD      DD  SS      SS      II      NN      NN  HH      HH
MM      MM      TT          HH      HH  DDDDDDDD  SSSSSSSS  IIIIII  NN      NN  HH      HH
MM      MM      TT          HH      HH  DDDDDDDD  SSSSSSSS  IIIIII  NN      NN  HH      HH

```



```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```



```

.....
.....
.....
.....

```


(2) 50
(3) 84
(4) 148

HISTORY ; Detailed Current Edit History
DECLARATIONS ; Declarative Part of Module
MT\$DSINH - Standard Double Precision Floating DSINH

MTH\$DSINH
1-008

K 15
; Double Floating Hyperbolic Sine routin 16-SEP-1984 01:21:28 VAX/VMS Macro V04-00
6-SEP-1984 11:22:46 [MTHRTL.SRC]MTHDSINH.MAR;1

Page 1
(1)

```
0000 1      .TITLE MTH$DSINH      ; Double Floating Hyperbolic Sine routine
0000 2      ; (DSINH)
0000 3      .IDENT /1-008/      ; File: MTHDSINH.MAR Edit: RNH1008
0000 4      ;
0000 5      ;*****
0000 6      ;
0000 7      ; COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      ; ALL RIGHTS RESERVED.
0000 10     ;
0000 11     ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     ; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     ; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     ; OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     ; TRANSFERRED.
0000 17     ;
0000 18     ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     ; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     ; CORPORATION.
0000 21     ;
0000 22     ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     ;
0000 25     ;
0000 26     ;*****
0000 27     ;
0000 28     ;
0000 29     ; FACILITY: MATH LIBRARY
0000 30     ;++
0000 31     ; ABSTRACT:
0000 32     ;
0000 33     ; MTH$DSINH is a function which returns the double floating hyperbolic sine
0000 34     ; of its double precision floating point argument. The call is standard
0000 35     ; call-by-reference.
0000 36     ;
0000 37     ;--
0000 38     ;
0000 39     ; VERSION: 01
0000 40     ;
0000 41     ; HISTORY:
0000 42     ; AUTHOR:
0000 43     ; Peter Yuo, 29-Jun-77: Version 01
0000 44     ;
0000 45     ; MODIFIED BY:
0000 46     ;
0000 47     ;
0000 48     ;
```



```
0000 50      .SBTTL HISTORY ; Detailed Current Edit History
0000 51
0000 52
0000 53 : ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000 54
0000 55 : Edit History for Version 01 of MTH$DSINH
0000 56
0000 57 : 0-2 MTH$$ERROR changed to MTH$$SIGNAL.
0000 58 : MTH$ ... changed to MTH
0000 59 : Changed error handling mechanism. Put error result in R0:R1 before
0000 60 : calling MTH$$SIGNAL in order to allow user modify error result.
0000 61
0000 62 : 0-3 Six term Taylor series, in powers of argument, replaced
0000 63 : by six term Chebyshev series, in powers of ARG**2,
0000 64 : with overhang provided. This improves accuracy. 18-May-1978;
0000 65 : Mary Payne.
0000 66 : 0-4 Fix LOG(2) constant. TNH 16-June-78
0000 67 : 1-001 - Update version number and copyright notice. JBS 16-NOV-78
0000 68 : 1-002 - Change MTH_FLOOVEMAT to MTH$K_FLOOVEMAT. JBS 07-DEC-78
0000 69 : 1-003 - Remove $$RMDEF - not needed. JBS 16-DEC-78
0000 70 : 1-004 - Add "" to the PSECT directive. JBS 22-DEC-78
0000 71 : 1-005 - Declare externals. SBL 17-May-1979
0000 72 : 1-006 - Use MTH$DEXP_R6. SBL 27-Sept-1979
0000 73 : 1-007 - Changed lower limit for Chebyshev approximation from 2**-27 to
0000 74 : 2**-28.
0000 75 : - Eliminated second call to EXP for input values between 28.5*ln2
0000 76 : and 127*ln2.
0000 77 : - Changed all final floating point divisions by 2 to interger
0000 78 : subtracts of 1 from the exponent field.
0000 79 : - Extended maximum range from 87.69 to 128*ln2=88.72.
0000 80 : - Changed logic for computing EXP(ixi-ln2) to reduce error.
0000 81 : - RNH 10-FEB-81
0000 82 : 1-008 - Change W^ to G^ on call to MTH$$SIGNAL RNH 09-Sept-1981
```

```
0000 84      .SBTTL  DECLARATIONS      ; Declarative Part of Module
0000 85
0000 86 :
0000 87 : INCLUDE FILES:      MTHJACKET.MAR
0000 88 :
0000 89 :
0000 90 :
0000 91 : EXTERNAL SYMBOLS:
0000 92 :
0000 93 :      .DSABL  GBL
0000 94 :      .EXTRN  MTH$DEXP R6
0000 95 :      .EXTRN  MTH$K_FLOOVEMAT
0000 96 :      .EXTRN  MTH$$SIGNAL
0000 97 :
0000 98 :
0000 99 : EQUATED SYMBOLS:
0000 100
0000 101      SD_1.0 = ^F1.0      ; 1.0
0000 102      value = 4      ; value.rd.r
0000 103
0000 104 :
0000 105 : MACROS:      none
0000 106 :
0000 107 :
0000 108 : PSECT DECLARATIONS:
0000 109
0000 110      .PSECT  _MTH$CODE      PIC,SHR,LONG,EXE,NOWRT
0000 111      ; program section for math routines
0000 112 :
0000 113 : OWN STORAGE:  none
0000 114 :
0000 115 :
0000 116 : CONSTANTS:
0000 117 :
0000 118 :
0000 119 D_127_LOG_2:
0000 120      .QUAD  ^X2BDAC7E20F3343B0      ; 127*ln2
0008 121 D_128_LOG_2:
0008 122      .QUAD  ^XCF78F7D1721743B1      ; 128*ln2-2**-49
0010 123 D_28.5_LOG_2:
0010 124      .QUAD  ^XDCC858B60990429E      ; 28.5*ln2
0018 125 D_LOG_2_HI:
0018 126      .QUAD  ^XCF80F7D172174031      ; (high 49 bits of ln2)+2**-49
0020 127 D_LOG_2_LO:
0020 128      .QUAD  ^XFF81898C86C3A5CA      ; ln2 - D_LOG_2_HI
0028 129
0028 130 DSINHTAB:
0028 131
0028 132      .WORD  ^0031727,^0072153
0028 133      .WORD  ^0121466,^0073017      ; DECIMAL:  0.2508223608819151D-07
0030 134      .WORD  ^0033470,^0167423
0034 135      .WORD  ^0154465,^0062340      ; DECIMAL:  0.2755729803646086D-05
0038 136      .WORD  ^0035120,^0006400
003C 137      .WORD  ^0150502,^0041641      ; DECIMAL:  0.1984126984813681D-03
0040 138      .WORD  ^0036410,^0104210
0044 139      .WORD  ^0104210,^0073152      ; DECIMAL:  0.8333333333332327D-02
0048 140      .WORD  ^0037452,^0125252
```



```

N 15
; Double Floating Hyperbolic Sine routin 16-SEP-1984 01:21:28 VAX/VMS Macro V04-00
DECLARATIONS ; Declarative Part of Modul 6-SEP-1984 11:22:46 [MTHRTL.SRC]MTHDSINH.MAR;1

```

Page 4
(3)

AAAD AAAA	004C	141	.WORD	^0125252,^0125255	; DECIMAL: 0.1666666666666667D+00
0000 0000	0050	142	.WORD	0,0	
0000 0000	0054	143	.WORD	0,0	; DECIMAL: 0.D0
	0058	144			
00000006	0058	145	DSINHLEN	= .- DSINH TAB/8	
	0058	146			

B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I


```
0058 148 .SBTTL MTH$DSINH - Standard Double Precision Floating DSINH
0058 149
0058 150
0058 151 ;++
0058 152 ; FUNCTIONAL DESCRIPTION:
0058 153
0058 154 ; DSINH - double precision floating point function
0058 155
0058 156 ; DSINH(X) is computed as:
0058 157
0058 158 ; If |X| < 2**(-28), DSINH(X) = X.
0058 159 ; If 2**(-28) <= |X| < 0.25, DSINH(X) = Chebyshev Series.
0058 160 ; If 0.25 <= |X| < 28.5*ln2, DSINH(X) = (DEXP(X) - DEXP(-X))/2.
0058 161 ; If 28.5 <= |X| < 127*ln2, DSINH(X) = sign(X)*EXP(|X|)/2.
0058 162 ; If 127*ln2 <= |X| < 128*ln2, then DSINH(X) = sign(X)*DEXP(|X|-ln2).
0058 163 ; If 128*ln2 < |X|, then overflow.
0058 164
0058 165 ; CALLING SEQUENCE:
0058 166
0058 167 ; DSINH.wd.v = MTH$DSINH(x.rd.r)
0058 168
0058 169 ; INPUT PARAMETERS:
0058 170
00000004 0058 171 ; LONG = 4 ; define longword multiplier
00000004 0058 172 ; x = 1 * LONG ; Contents of x is the argument
0058 173
0058 174 ; IMPLICIT INPUTS: none
0058 175
0058 176 ; OUTPUT PARAMETERS:
0058 177
0058 178 ; VALUE: double precision floating hyperbolic sine of the argument
0058 179
0058 180 ; IMPLICIT OUTPUTS: none
0058 181
0058 182 ; COMPLETION CODES: none
0058 183
0058 184 ; SIDE EFFECTS:
0058 185
0058 186 ; Signal: MTH$ FLOOVEMAT if 128*ln2 <= |X| with reserved operand in R0/R1
0058 187 ; (copied to the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
0058 188 ; Associated message is: 'FLOATING OVERFLOW IN MATH LIBRARY'. Result is reserved
0058 189 ; operand -0.0 unless a user supplied (or any) error handler changes CHF$MCH_R0/R1
0058 190
0058 191 ; NOTE: This procedure disables floating point underflow, enables integer
0058 192 ; overflow.
0058 193
0058 194 ; ---
0058 195
0058 196
0058 197 .ENTRY MTH$DSINH, ^M<IV, R2, R3, R4, R5, R6, R7>
005A 198 ; standard call-by-reference entry
005A 199 ; disable DV (and FU), enable IV
005A 200 MTH$FLAG_JACKET ; flag that this is a jacket procedure in
005A
005A MOVAB G^MTH$$JACKET_HND, (FP) ; set handler address to jacket
0061 ; handler
0061
```

6D 00000000'GF 9E


```

0061 201
0061 202
0061 203
56 04 BC 70 0061 204 MOVD @value(AP), R6
50 50 56 7D 0065 205 MOVQ R6, R0
50 8000 8F AA 0068 206 BICW2 #^X8000, R0
3F80 8F 50 B1 006D 207 CMPW R0, #^X3F80
1A 18 0072 208 BGEQ GEQ_TO_0.25
0074 209
0074 210
0074 211 : |X| < 0.25
0074 212 :
0074 213
3280 8F 50 B1 0074 214 CMPW R0, #^X3280
04 18 0079 215 BGEQ GEQ_TO_2M28
007B 216
007B 217 : |X| < 2**-28
007B 218 :
007B 219 :
50 56 7D 007B 220 MOVQ R6, R0
04 04 007E 221 RET
007F 222
007F 223
007F 224 : 2**-28 =< |X| < 0.25
007F 225 :
007F 226 :
007F 227
007F 228 GEQ_TO_2M28:
A1 AF 50 50 64 007F 229 MULD R0, R0
05 50 75 0082 230 POLYD R0, #DSINHLEN-1, DSINHTAB
0087 231
0087 232
50 56 64 0087 233 MULD R6, R0
50 56 60 008A 234 ADDD R6, R0
04 008D 235 RET
008E 236
008E 237 : 0.25 =< |X|
008E 238 :
008E 239 :
008E 240
008E 241 GEQ_TO_0.25:
FF6D CF 50 71 008E 242 CMPD R0, D 127 LOG_2
3B 14 0093 243 BGTR GTR_TRAN_T27_LOG_2
0095 244
0095 245 : 0.25 =< |X| < 127*ln2
0095 246 :
0095 247 :
0095 248
FF76 CF 50 71 0095 249 CMPD R0, D 28.5 LOG_2
20 14 009A 250 BGTR ONLY_ONE_TERM
009C 251
009C 252 : 0.25 =< |X| < 28.5*ln2
009C 253 :
009C 254 :
50 56 7D 009C 255 MOVQ R6, R0
00000000*EF 16 009F 256 JSB MTH$DEXP_R6

```

; case of an error in routine
; If an error, convert signal to user PC
; and resignal
; R6/R7 = |X| = @value(AP)

; R0/R1 = |X|
; compare |X| with 0.25
; branch if |X| >= 0.25

; compare |X| with 2**-28
; branch if |X| >= 2**-28

; R0/R1 = X
; return with result = argument

; Get ARG**2 for POLYD
; R0/R1 = SUM(Ci*X**i), with
; Last coefficient zero
; MULD by ARG, and then
; Add in ARG with overhang.
; return with result in R0/R1

; compare |X| with 127*ln2
; branch if |X| > 127*ln2

; Compare |X| with 28.5*ln2, if
; greater, only one of EXP(X) and
; EXP(-X) is significant.

; R0/R1 = X
; R0/R1 = DEXP(X)


```

      7E 50 7D 00A5 257      MOVQ    R0, -(SP)      ; push DEXP(X) on stack
      50 04 BC 72 00A8 258      MNEGD   @value(AP), R0      ; R0/R1 = -X
      00000000'EF 16 00AC 259      JSB     MTH$DEXP R6      ; R0/R1 = DEXP(-X)
      50 8E 50 63 00B2 260      SUBD3   R0, (SP)+, R0      ; R0/R1 = DEXP(X) - DEXP(-X)
      50 0080 8F A2 00B6 261      SUBW    #^X0080, R0      ; R0/R1 = (DEXP(X)-DEXP(-X))/2
      04 00BB 262      RET                                ; return with result in R0/R1
      00BC 263
      00BC 264      ; 28.5*ln2 =< |X| < 127*ln2
      00BC 265
      00BC 266
      00BC 267
      00BC 268      ONLY_ONE_TERM:
      00000000'EF 16 00BC 269      JSB     MTH$DEXP R6      ; R0/R1 = DEXP(|X|)
      04 BC 73 00C2 270      TSTD     @value(AP)      ; Check sign of X
      03 14 00C5 271      BGTR     POSITIVE      ; If negative change sign of
      50 50 72 00C7 272      MNEGD   R0, R0      ; DEXP(|X|)
      00CA 273      POSITIVE:
      50 0080 8F A2 00CA 274      SUBW    #^X0080, R0      ; R0/R1 = sign(X)*DEXP(|X|)/2
      04 00CF 275      RET
      00D0 276
      00D0 277      ; 127*ln2 < |X|
      00D0 278
      00D0 279
      00D0 280      GTR_THAN_127_LOG_2:
      FF33 CF 50 71 00D0 281      CMPD    R0, D_128_LOG_2      ; Compare |X| to 128*ln2
      1D 18 00D5 282      BGEQ     ERROR      ; if 128*ln2 =< |X|, overflow occurs
      00D7 283
      00D7 284
      00D7 285
      00D7 286      ; 127*ln2 < |X| < 128*ln2
      00D7 287
      00D7 288
      50 FF3D CF 62 00D7 289      SUBD    D_LOG_2_HI, R0      ; R0/R1=|X|-D_LOG_2_HI
      00000000'EF 16 00DC 290      JSB     MTH$DEXP R6      ; R0/R1 = DEXP(|X|-D_LOG_2_HI)
      52 50 FF3A CF 65 00E2 291      MULD3   D_LOG_2_LO, R0, R2      ; R2/R3=DEXP(|X|-D_LOG_2_HI)*D_LOG_2_LO
      50 52 62 00E8 292      SUBD    R2, R0      ; R0/R1=DEXP(|X|-ln2)
      04 BC 73 00EB 293      TSTD     @value(AP)      ; test the sign of X
      03 18 00EE 294      BGEQ     10$      ; branch if X >= 0
      50 50 72 00F0 295      MNEGD   R0, R0      ; R0/R1 = sign(X) * DEXP(|X|-ln2)
      04 00F3 296      10$: RET      ; return with result in R0/R1
      00F4 297
      00F4 298
      00F4 299      ; 128*ln2 =< |X|, error
      00F4 300
      00F4 301
      7E 00'8F 9A 00F4 302      ERROR: MOVZBL #MTH$K_FLOOVEMAT, -(SP) ; condition value
      50 01 0F 79 00F8 303      ASHQ     #15, #T, R0      ; R0/R1 = result = reserved operand -0.0
      00FC 304
      00FC 305      ; goes to signal mechanism vector
      00FC 306      ; (CHF$MCH_R0/R1) so error handler
      00FC 307      ; can modify the result.
      00000000'GF 01 FB 00FC 307      CALLS   #1, G^MTH$$SIGNAL      ; signal error and use real user's PC
      04 0103 308      RET      ; independent of CALL vs JSB
      0104 309
      0104 310
      0104 311
      0104 312
      0104 313      .END      ; return - R0/R1 restored from CHF$MCH_R0/
```


MTH\$DSINH
Symbol table

E 16
; Double Floating Hyperbolic Sine routin 16-SEP-1984 01:21:28 VAX/VMS Macro V04-00
6-SEP-1984 11:22:46 [MTHRTL.SRC]MTHDSINH.MAR;1

Page 8
(4)

```
DSINHLEN      = 00000006
DSINHNTAB     = 00000028 R      01
D_127_LOG_2   = 00000000 R      01
D_128_LOG_2   = 00000008 R      01
D_28.5_LOG_2  = 00000010 R      01
D_LOG_2_HI    = 00000018 R      01
D_LOG_2_LO    = 00000020 R      01
ERROR         = 000000F4 R      01
GEQ_TO_0.25   = 0000008E R      01
GEQ_TO_2M28   = 0000007F R      01
GTR_THAN_127_LOG_2 = 000000D0 R      01
LONG          = 00000004
MTH$$JACKET_HND ***** X      01
MTH$$SIGNAL   ***** X      00
MTH$DEXP_R6   ***** X      00
MTH$DSINH     00000058 RG     01
MTH$K_FLOOVEMAT ***** X      00
ONLY_ONE_TERM 000000BC R      01
POSITIVE      000000CA R      01
VALUE        = 00000004
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
_MTH\$CODE	00000104 (260.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG				

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.12	00:00:01.77
Command processing	120	00:00:00.67	00:00:05.67
Pass 1	88	00:00:00.90	00:00:05.83
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	68	00:00:00.71	00:00:03.50
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	319	00:00:02.46	00:00:16.83

The working set limit was 900 pages.
4682 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 22 non-local and 1 local symbols.
373 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_S255\$DUA28:[SYSLIB]STARLET.MLB;2

0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:MTHDSINH/OBJ=OBJ\$:MTHDSINH MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY